

# 游易-程序主题分享合集

[+ 收藏专题](#)

知识管理部 等 2022.06.07 15:37

6242

321

184个资源

汇总从101至今的游易征稿文章合集。

[推荐资源](#) [站内分享](#) [用手机查看](#) [引用](#) [投稿](#) [© 分享至POPO眼界大开](#)
[专题首页](#) > [110游戏开发工具](#) > [Windows C++ 程序员三板斧](#)

## Windows C++ 程序员三板斧



汤泽江

2017.01.10 17:02

2388

17

8

[查看原文](#)

本文仅面向以下用户开放，请注意内容保密范围

查看权限：互娱正式-公开

“项目规模越来越大，C++ 程序员会头痛两个问题：内存和性能。有啥神器，来查内存/性能问题？”

### 一板斧：tcmalloc

我们都知道 google perftools 中有个 tcmalloc，可以极大提高 malloc/free 的效率。

tcmalloc 其实还自带了一个 heap profiler，可以方便观察内存都来自于哪些代码。

官方仓库：<https://github.com/gperftools/gperftools>

鉴于官方版本对win32支持不太好，我自己维护了一个win32 static-build版本，定期同步官方修改。

<https://github.com/kasicass/gperftools-win32>

修正了官方版本 callstack 抓不准、heap profiler 会卡住等等问题。

win32 程序只需要 static link 此版本的 .lib，就能自动启用 tcmalloc（性能提升），并且很方便可以开启 heap profiler。

可以参考自带例子 libtcmalloc-static-test。

libtcmalloc-static-test.exe 运行一下，D: 上得到4个 .heap 文件。

gperftools-win32\tools 下有两个小工具，PrintHeapMax.py 和 PrintHeapLeak.py，分别用于查内存占用情况和内存泄漏情况。

比如：

```
> python PrintHeapMax.py --limit 0 d:\heap.0003.heap
liveCount: 1, liveBytes: 0.00 Mb, sumCount: 1, sumBytes: 0.00 Mb
__tmainCRTStartup(f:\dd\vctools\crt_bld\self_x86\crt\src\crt0.c:241)
BaseThreadInitThunk(???)
RtlSubscribeWnfStateChangeNotification(???)
RtlSubscribeWnfStateChangeNotification(???)

liveCount: 0, liveBytes: 0.00 Mb, sumCount: 4, sumBytes: 0.01 Mb
BaseThreadInitThunk(???)
RtlSubscribeWnfStateChangeNotification(???)
RtlSubscribeWnfStateChangeNotification(???)
```

PrintHeapMax.py 打印出哪些地方使用了内存，降序排列。

--limit N，表示打印出“当前 >= N Mb”的内存占用

liveCount, 分配次数

liveBytes, 当前占用内存数量

sumCount, 峰值分配次数

sumBytes, 峰值占用内存数量

PrintHeapLeak.py 打印出 liveCount == sumCount 的情况  
-limit N, 表示打印出"当前 >= N Mb"的内存占用

count 就是 liveCount, 也是sumCount。  
liveCount 等于 sumCount, 我们认为从来没释放, 可能存在泄漏。

如何将 heap profiler 用于实际项目?

可以将 heap profiler 作为引擎的一个开关, QA发现内存异常, 就打开开关, 启动游戏, 引擎内存每上涨100M, 就自动 dump 一个 .heap。QA跑一轮游戏, 将最后两三个 .heap 发给程序分析即可。

QA发给我们的 .heap, 需要手工修改 MAPPED\_LIBRARIES 下的第一行, 将QA中的.exe路径, 改为本机带了.pdb的.exe路径。之后就可以正确分析了。

MAPPED\_LIBRARIES:

00870000-008df000 r-xp 00000000 00:00 0 path\to\game.exe

天下3中一个实际例子

```
liveCount: 118736, liveBytes: 113.89 Mb, sumCount: 118736, sumBytes: 113.89 Mb
anonymous namespace':LibcInfoWithPatchFunctions<3>::~PerfTools_deletearray(c:\mf\src\lib\python\objects\obmalloc.c:881)
new_arena(c:\mf\src\lib\python\objects\obmalloc.c:881)
new_arena(??:0)
insertdict_by_entry(c:\mf\src\lib\python\objects\dictobject.c:519)
all_contained_in(c:\mf\src\lib\python\objects\dictobject.c:2827)
PyEval_EvalFrameEx(c:\mf\src\lib\python\python\ceval.c:2243)
PyEval_EvalFrameEx(c:\mf\src\lib\python\python\ceval.c:1247)
PyEval_EvalCodeEx(c:\mf\src\lib\python\python\ceval.c:3471)
PyImport_ImportFrozenModule(c:\mf\src\lib\python\python\import.c:2098)
my_ExecCodeModuleEx(c:\mf\src\lib\python\python\import.c:3741)
load_next(c:\mf\src\lib\python\python\import.c:2538)
is_builtin(c:\mf\src\lib\python\python\import.c:1199)
find_module(c:\mf\src\lib\python\python\import.c:1336)
init_builtin(c:\mf\src\lib\python\python\import.c:2019)
```

## 二板斧: verysleepy

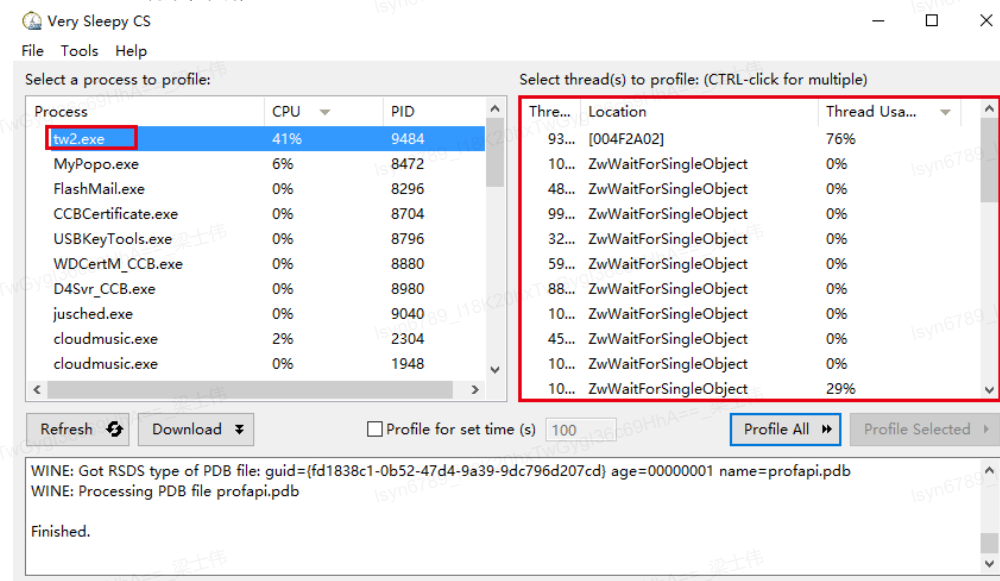
<http://www.codersnotes.com/sleepy/>

verysleepy 是用于查函数消耗的工具, 方便实用, 强力推荐!

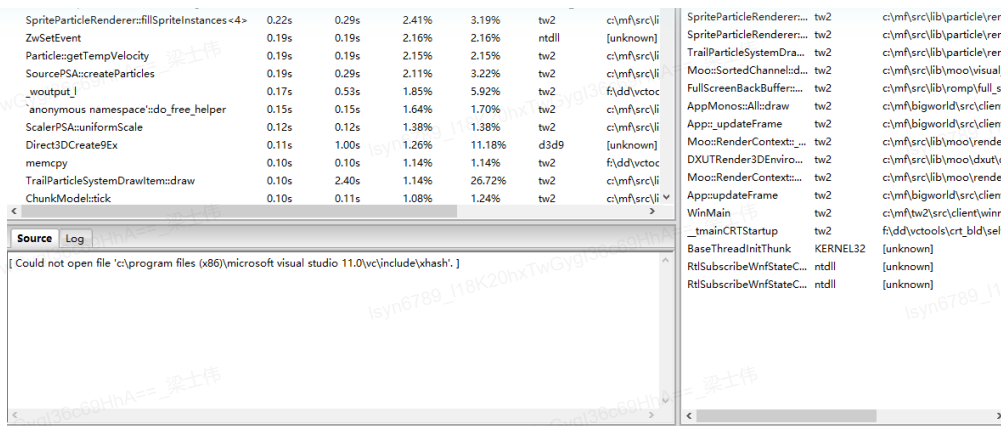
左边框里选进程, 右边框里选线程。

Profile All 统计整个进程 (所有线程)。

Profile Selected 统计单个线程。



只需要配上 .pdb, 可以很方便看到每个函数的消耗, 调用栈等等。



对于游戏中偶尔的卡顿，又能重现的，可以用 versleepy 抓一下，看看消耗在哪儿。

### 三板斧：Windows Performance Analysis (WPA)

查性能还有一个神器，微软自家产的WPA。

WPA功能强大，但用起来比较复杂，这里提供一些学习资料。

<http://randomascii.wordpress.com/2014/08/19/etw-training-videos-available-now/>

Windows Performance Series

<http://channel9.msdn.com/Series/Windows-Performance>

Windows Performance Toolkit Technical Reference

<http://msdn.microsoft.com/en-us/library/windows/apps/hh162945.aspx>

UIforETW

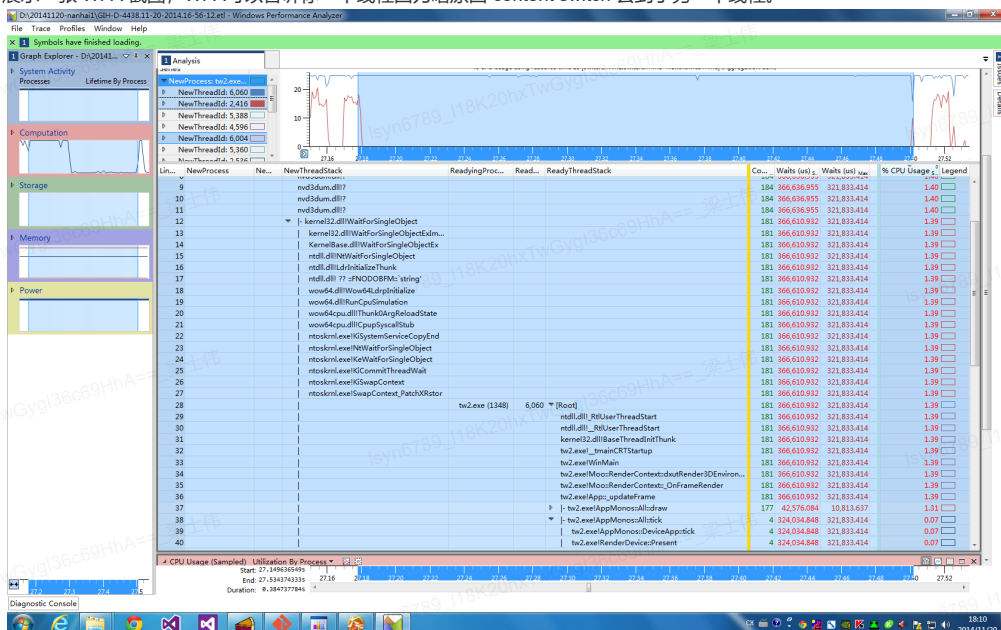
<https://github.com/google/UIforETW>

<https://randomascii.wordpress.com/2015/04/14/uiforetw-windows-performance-made-easier/>

ETW Heap Tracing -- Every Allocation Recorded

<https://randomascii.wordpress.com/2015/04/27/etw-heap-tracingevery-allocation-recorded/>

展示一张 WPA 截图，WPA可以告诉你一个线程因为啥原因 content switch 去到了另一个线程。



本内容仅代表个人观点，不代表网易游戏，仅供内部分享传播，不允许以任何形式外泄，否则追究法律责任。



快来成为第一个打赏的人吧~

### 全部评论 8



请输入评论内容

还可以输入 500 个字



(可添加1个视频+5张图片)

☐ 匿名

评论

最热

最新



匿名

8楼



2017-12-25 17:33

回复

0



匿名

7楼



2017-08-07 19:31

回复

0



Wade(吴建江)

6楼

233333

2017-07-17 10:34

回复

0



匿名

5楼



2017-03-28 10:28

回复

0



匿名

4楼



2017-03-16 13:42

回复

0



王子一

3楼



2017-03-14 19:39

回复

0

首页

专题

职业库

易播

现场教学

游戏资讯

乐问

搜全站



职业库

112MOBA技术

111画质与效率

110游戏开发工具

109-设计模式

108产品计费

107偏底层&小而美

106-物理引擎

105端游和手游



段兆晔

1楼 太赞了!!! 收藏

2017-02-28 17:11

回复 0

加载完毕,没有更多了



Share us  
your growing

常用链接

易协作  
OA

会议预定  
文具预定

游戏部IT资源  
易网

网易POPO  
工作报告



POPO服务号



KM APP下载

平台用户协议 帮助中心

新闻